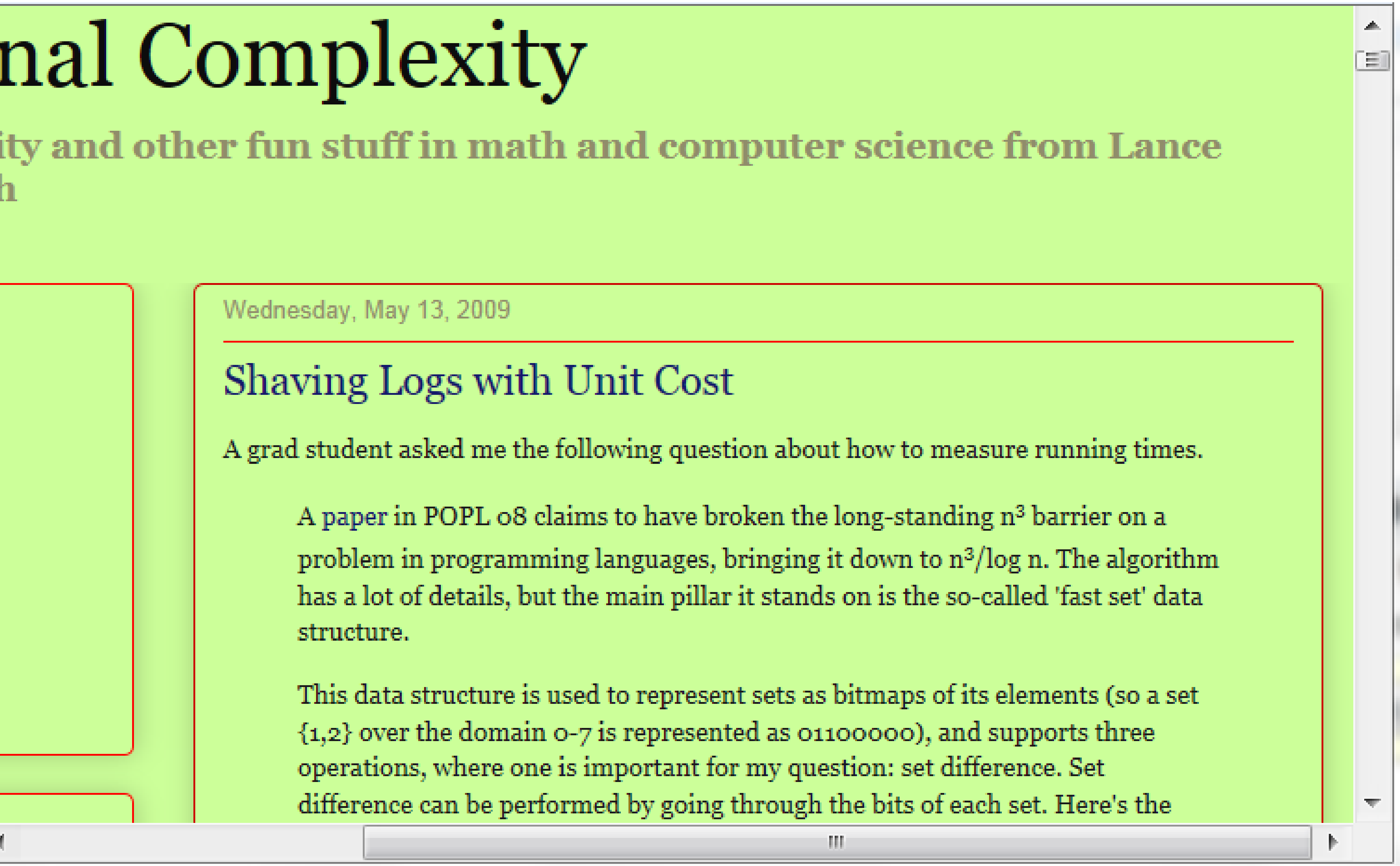
The Art of Shaving Logs

Timothy Chan
U. of Waterloo

(from http://en.wikipedia.org/wiki/The_Art_of_Shaving)



nal Complexity

ity and other fun stuff in math and computer science from Lance
h

Wednesday, May 13, 2009

Shaving Logs with Unit Cost

A grad student asked me the following question about how to measure running times.

> A paper in POPL 08 claims to have broken the long-standing n^3 barrier on a problem in programming languages, bringing it down to $n^3/\log n$. The algorithm has a lot of details, but the main pillar it stands on is the so-called 'fast set' data structure.
>
> This data structure is used to represent sets as bitmaps of its elements (so a set {1,2} over the domain 0-7 is represented as 01100000), and supports three operations, where one is important for my question: set difference. Set difference can be performed by going through the bits of each set. Here's the

(from http://blog.computationalcomplexity.org/2009/05/shaving-logs-with-unit-cost.html)

Theme

$$O(n^a) \rightarrow O(n^a / \log^b n)$$

Examples

- Boolean matrix multiplication in $O(n^3 / \log^2 n)$ time [Arlazarov,Dinic,Kronrod,Faradzev'70 — the "4 Russians"]
- All-pairs shortest paths of real-weighted graphs & min-plus matrix multiplication in $O(n^3 \log^3 \log n / \log^2 n)$ time [Fredman,FOCS'75, …, C.,WADS'05, …, C.,STOC'07]
- LCS & edit distance for bounded alphabet in $O(n^2 / \log n)$ time [Masek,Paterson'80]
- Maximum unweighted bipartite matching in $O(n^{5/2} / \log n)$ time [Alt,Blum,Mehlhorn,Paul'91, Feder,Motwani,STOC'91]
- Regular expression matching in $O(nP / \log n)$ time [Myer'92]
- 3SUM in $O(n^2 \log^2 \log n / \log^2 n)$ time [Baran,Demaine, Pătraşcu,WADS'05]
- Transitive closure for sparse graphs in $O(mn / \log n)$ time
- All-pairs shortest paths for sparse unweighted undirected graphs in $O(mn / \log n)$ time (for m >> n log n) [C.,SODA'06]

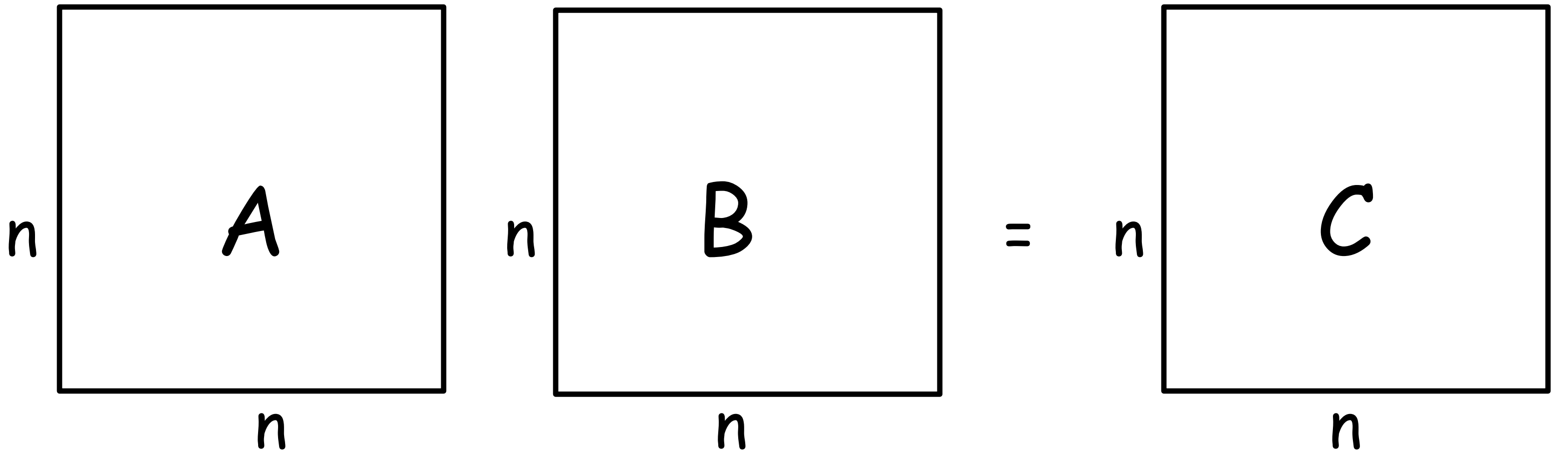
Examples (Cont'd)

- Min-plus convolution in $O(n^2 \log^3 \log n / \log^2 n)$ time [Bremner,C.,Demaine,Erickson,Hurtado,Iacono,Langerman,Pătraşcu, Taslakian,ESA'06]
- CFL reachability in $O(n^3 / \log^2 n)$ or $O(mn/\log n)$ [Chaudhuri'08]
- k-cliques in $O(n^k / \log^{k-1} n)$ time [Vassilevska'09]
- Diameter of real-weighted planar graphs in $O(n^2 \log^4 \log n / \log n)$ time [Wulff-Nilsen'10]
- Discrete Fréchet distance decision in $O(n^2 \log\log n / \log n)$ time [Agarwal,Avraham,Kaplan,Sharir,SODA'13]
- Continuous Fréchet distance decision in $O(n^2 \log^2 \log n/\log n)$ time [Buchin,Buchin,Meulemans,Mulzer'12 — "4 Soviets walk the dog"]
- Klee's measure problem in $O(n^{d/2} \log^{O(1)} \log n / \log^{d/2-2} n)$ time [C.,FOCS'13]
- Etc. etc. etc.

PART 1:

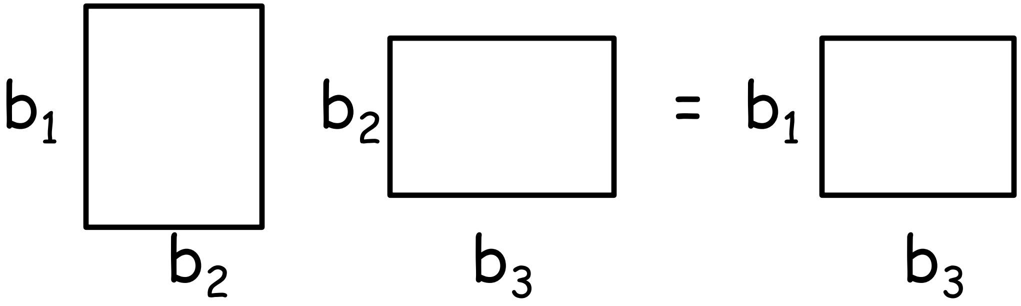
Unweighted Problems

Example 1.1: Boolean Matrix Multiplication



Example 1.1: Boolean Matrix Multiplication First Alg'm

- $T(n) \leq (n/b_1)(n/b_2)(n/b_3)\,[T(b_1,b_2,b_3) \;\; + O(b_1 b_3)]$

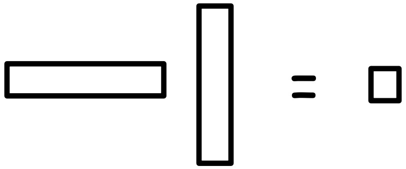


Example 1.1: Boolean Matrix Multiplication First Alg'm

- $T(n) \le (n/b_1)(n/b_2)(n/b_3)\,[T(b_1,b_2,b_3) \;\; + O(b_1 b_3)]$

- Notation: w = machine word size

- $T(1,w,1) = O(1)$
  - by one word op (bitwise-&)



$$\Rightarrow \; T(n) = O(n\cdot(n/w)\cdot n) \le \boxed{O(n^3 / \log n)}$$

Standard RAM Model

- $w \geq \log n$ (pointers/indices fit in a word)
- Unit cost for standard (arithmetic, bitwise-logical, shift) ops on words

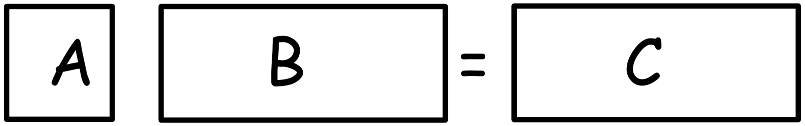
Example 1.1: Boolean Matrix Multiplication
Second Alg'm [Arlazarov,Dinic,Kronrod, Faradzev'70]: "4 Russians"

- $T(n) \leq (n/b_1)(n/b_2)(n/b_3)\,[T(b_1,b_2,b_3) \;\; + O(\,b_1 b_3/w)]$

by word ops (bitwise-or)

- Notation: $w_0 = \varepsilon \log n$

- $T(w_0,w_0,n) \;=\; O(n)$:
  - multiply A with all 2^{w_0} possible column vectors in time $O(2^{w_0}\; w_0^{\,2}) \;=\; n^{O(\varepsilon)}$
  - then do n table lookups



$$\Rightarrow \; T(n) \;=\; O((n/w_0)\cdot(n/w_0)\cdot 1 \;\cdot \cancel{w_0} n) \;=\; \boxed{O(n^3 / \log^2 n)}$$

Example 1.1: Boolean Matrix Multiplication Remarks

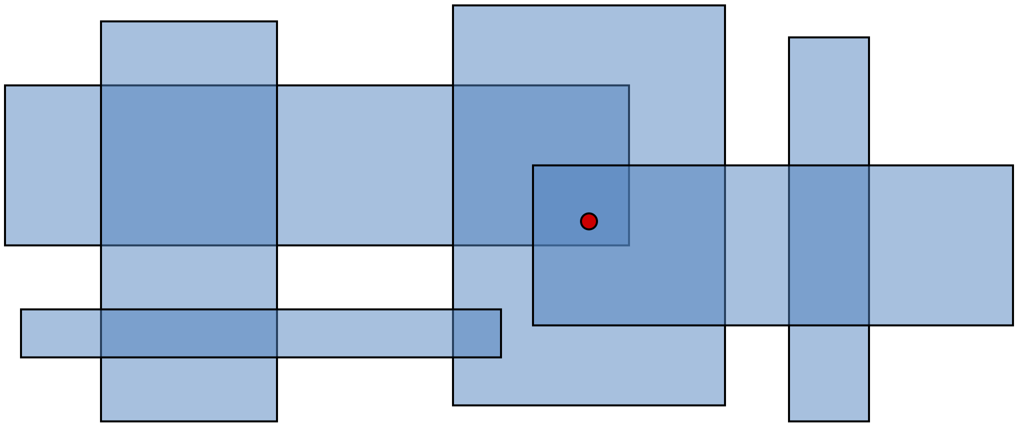
- For sparse matrices, $O(mn/\log n)$ time
- $O^*(n^3 / \log^{9/4} n)$ time [Bansal,Williams,FOCS'09]
- Of course, $O(n^{2.38})$ is still better (theoretically)

Are We Cheating?

- $w \geq \log n$ assumption is implicit in traditional alg'm analysis
- Basic principle of table lookup:
  - avoid solving same subproblem again!
- Word ops on words of size $w_0 = \varepsilon \log n$ can be simulated by table lookup
- Some alg'ms can even be re-implemented in pointer machine model

Example 1.2: Box Depth

- Given n boxes in d ≥ 3 dimensions,
  - find a point with max/min depth
    where depth(p) = # of boxes containing p



Example 1.2: Box Depth Alg'm [C.,SoCG'08/FOCS'13]

- $T(n) \leq O(n/b)^{d/2} [T(b) + O^*(b/w_0)]$
  - by comp. geometry techniques...
- For $b = w_0/\log w_0$, $T(b) = O(1)$:
  - encode input in $O(b \log b) = O(w_0)$ bits
  - precompute all answers in time $2^{O(w_0)} = n^{O(\varepsilon)}$
  - then do table lookup

$\Rightarrow$ $\boxed{O^*((n/\log n)^{d/2} \cancel{\log n})}$

- Notation: O^* hides loglog n factors

PART 2:

Integer-Valued Problems

Integer Word-RAM Model

- Input numbers are integers in $\{0,\dots,U\}$ $(U \geq n)$
- $w \geq \log U$ (input numbers fit in a word)
- Unit cost for standard ops on words

Example 2.1: 3SUM

- Given 3 sets of n numbers A, B, C,
  - do there exist a in A, b in B, c in C with a+b+c = 0?

Example 2.1: 3SUM Standard Alg'm

- Pre-sort A, B, C
- For each c in C:
  - test whether A+c and -B have a common element by linear scan

$\Rightarrow \boxed{O(n^2)}$ time

Example 2.1: 3SUM
An Alg'm by Baran,Demaine,Pătraşcu [WADS'05]

- Pre-sort A, B, C
- For each c in C:
  - test whether A+c and -B have a common element by linear scan:
    - hash, e.g., by taking mod random prime $p \sim w_0^{100}$ (test for $a+b+c = 0 \mod p$)

    ⇒ list has $O(n \log w_0)$ bits

    ⇒ linear scan takes $O^*(n/w_0)$ time

⇒ $\boxed{O^*(n^2 / \log n)}$ time (randomized)

Example 2.1: 3SUM
Remarks

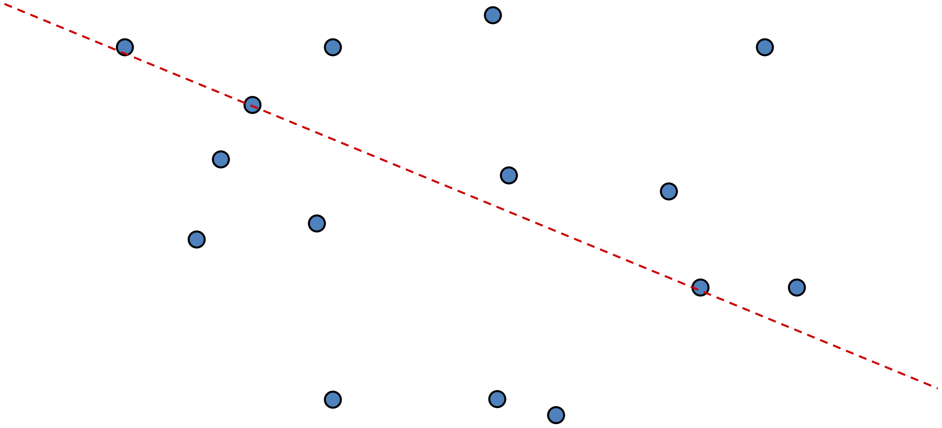
- Generalizes to asymmetric version with $|C| = m \leq n$ in $O^*(mn / \log n)$ time

- Another alg'm of Baran, Demaine, Pătraşcu in $O^*(n^2 / \log^2 n)$ time (randomized)

- Generalizes to kSUM problem in $O^*(n^{(k+1)/2} / \log n)$ time for odd k:
  - reduces to asymmetric 3SUM with $|A|=|B| = n^{(k-1)/2}$, $|C| = n$

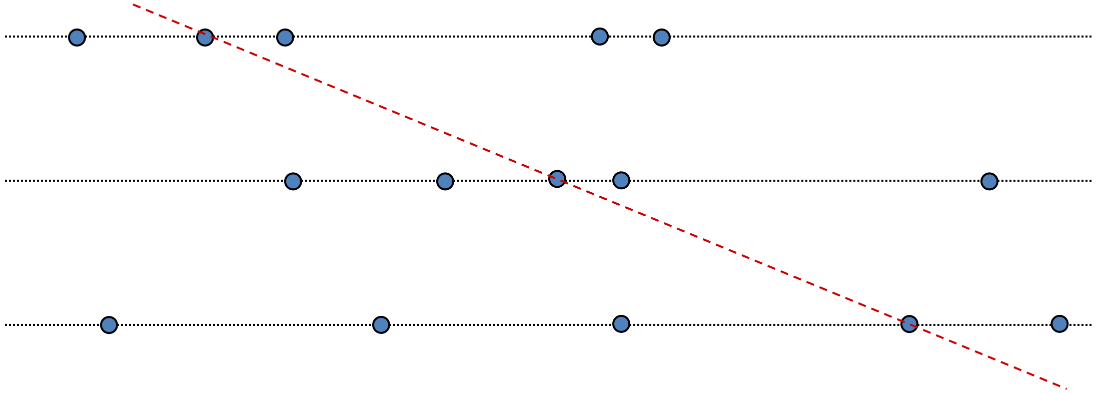
Example 2.2: 3-Collinearity in 2D

- Given n points in 2D,
  - do there exist 3 collinear points?



Example 2.2: 3-Collinearity in 2D

- Note: 3SUM reduces to 3-collinearity [Gajentaan,Overmars'95]

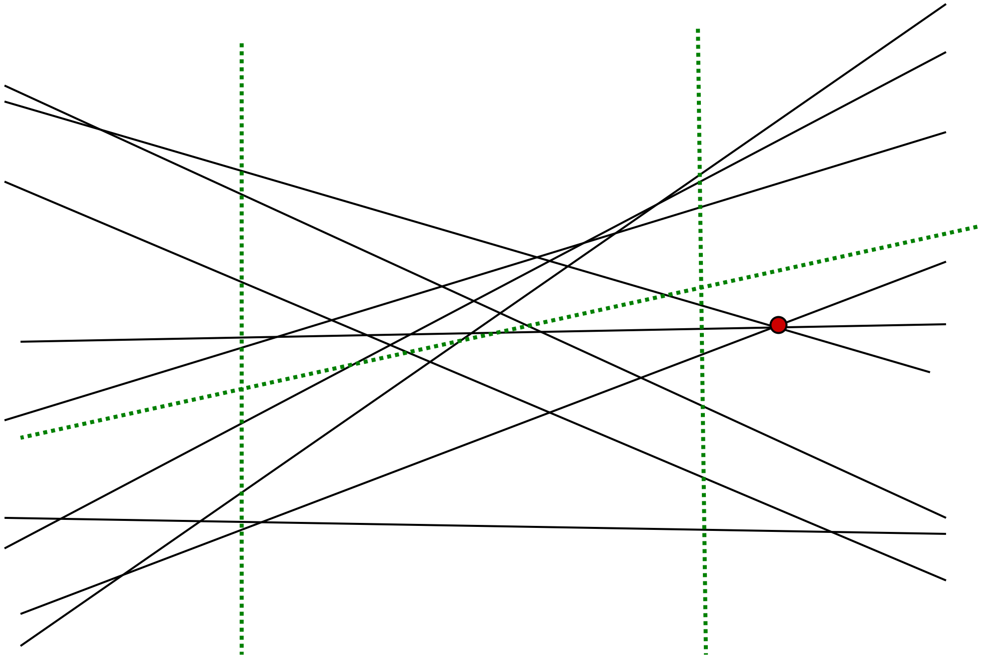


- Baran,Demaine,Pătraşcu asked: can 3-collinearity also be solved in $O(n^2 / \text{polylog } n)$ time for integer coords?

YES!

Example 2.2: 3-Collinearity in 2D Alg'm [C.,unpublished'06]

- $T(n) \leq O(r^2)\, T(n/r) + O(nr)$
  - by (1/r)-cuttings in the dual [Clarkson,Shor'89, Chazelle,Friedman'93]



Example 2.2: 3-Collinearity in 2D Alg'm [C.,unpublished'06]

- $T(n) \leq O(n/b)^2\, T(b) + O^*(n(n/b)/w_0) + O^*(n(n/b)^2/w_0^2)$
  - by (1/r)-cuttings in the dual [Clarkson,Shor'89, Chazelle,Friedman'93]

- For $b = w_0/\log w_0$, $T(b) = O(1)$:
  - hash coordinates by taking mod random prime $p \approx w_0^{100}$ (test for $(x_2-x_1)(y_3-y_1) = (y_2-y_1)(x_3-x_1) \mod p$)

  $\Rightarrow$ encode input in $O(b \log w_0) = O(w_0)$ bits
  - then do table lookup

$\Rightarrow$ $T(n) = \boxed{O^*(n^2 / \log^2 n)}$ (randomized)

Example 2.2: 3-Collinearity in 2D
Remarks

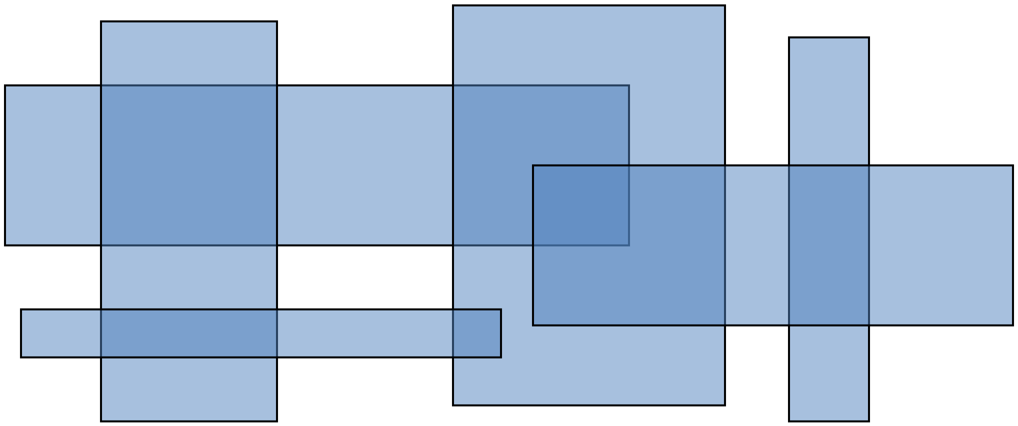
- Generalizes to affine degeneracy testing in d dimensions in $O^*(n^d / \log^d n)$ time

- But does not generalize to kSUM for larger k, or asymmetric 3SUM

- Open Question: other 3SUM-hard problems
  - e.g., 3 points with min triangle area??

Example 2.3: Klee's Measure Problem

- Given n boxes in d ≥ 3 dimensions,
  - find volume of the union of the boxes



Example 2.3: Klee's Measure Problem Alg'm [C.,FOCS'13]

- $T(n) \leq O(n/b)^{d/2} [T(b) + O(b)]$

- For $b = w_0/\log\log U$, $T(b) = O(\log U/\log\log U)$:
  - encode arrangement of boxes in $O(b \log b) \leq O(w_0)$ bits
  - hash coords by taking mod different primes $p \approx \log U$ (e.g., in 3D, volume has the form $\Sigma \pm x_i y_j z_k \mod p$)
    $\Rightarrow$ encode coordinates in $O(b \log\log U) = O(w_0)$ bits
  - # different primes = $O(\log U/\log\log U)$
  - reconstruct volume by Chinese remainder theorem !

$\Rightarrow T(n) = \boxed{O^*((n/\log n)^{d/2} \cancel{\log U})}$

$\log^2 n$ assuming $n > w$
(by more ideas)

PART 3:

Real-Valued Problems

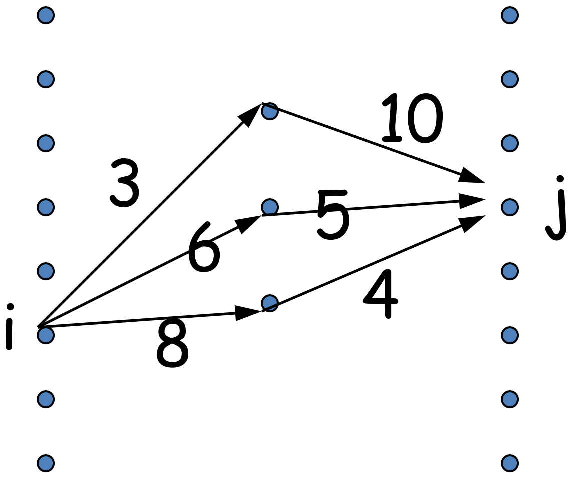
Real RAM Model

- Input numbers are reals

- Unit cost for standard arithmetic/comparison ops on reals, & for (log n)-bit pointers

Example 3.1: All-Pairs Shortest Paths & Min-Plus Matrix Multiplication

- Given n x n matrices A, B,
  - compute $c_{ij} = \min_k (a_{ik} + b_{kj})$



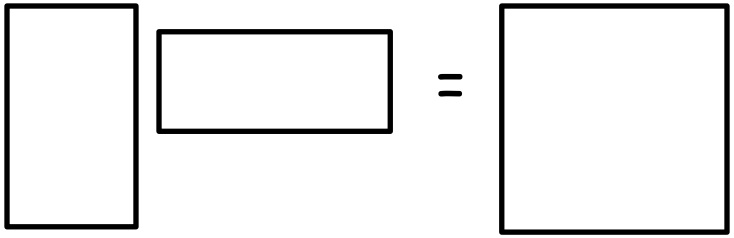
$c_{ij} = 11$

Example 3.1: All-Pairs Shortest Paths History

• Fredman [FOCS'75]	$O(n^3 \log^{1/3}\log n / \log^{1/3} n)$
• Takaoka'92	$O(n^3 \log^{1/2}\log n / \log^{1/2} n)$
• Dobosiewicz'90	$O(n^3 / \log^{1/2} n)$
• Han'04	$O(n^3 \log^{5/7}\log n / \log^{5/7} n)$
• Takaoka [COCOON'04]	$O(n^3 \log^{2}\log n / \log n)$
• Zwick [ISAAC'04]	$O(n^3 \log^{1/2}\log n / \log n)$
• Chan [WADS'05]	$O(n^3 / \log n)$
• Han [ESA'06]	$O(n^3 \log^{5/4}\log n / \log^{5/4} n)$
• Chan [STOC'07]	$O(n^3 \log^{3}\log n / \log^{2} n)$
• Han,Takaoka [SWAT'12]	$O(n^3 \log\log n / \log^{2} n)$

Example 3.1: All-Pairs Shortest Paths Decision Tree Complexity

(If We Only Count Comparisons...) [Fredman,FOCS'75]



- $T(n,n^{1/2},n) = O(n^2 \cancel{\log n})$:
  - idea: $a_{ik} + b_{kj} \le a_{ik'} + b_{k'j}$

    $\Leftrightarrow \quad a_{ik} - a_{ik'} \le b_{k'j} - b_{kj}$
  - n choices for i, j, $n^{1/2}$ choices for k, k'

  $\Rightarrow$ $O(n^2)$ values for left/right-hand side
  - sort all these values!

$\Rightarrow \quad T(n) = \boxed{O(n^{2.5} \cancel{\log n})}$

Example 3.1: All-Pairs Shortest Paths
An Alg'm by Fredman [FOCS'75]

- $T(n) \leq (n/b_1)(n/b_2)(n/b_3)\,[T(b_1,b_2,b_3) + O(b_1 b_3)]$

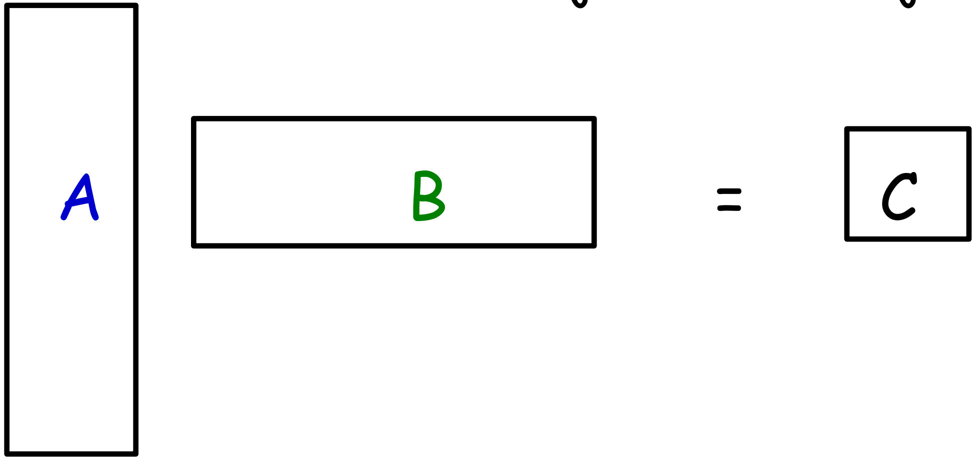
- For $b = w_0^{1/2}$, $T(b,b^{1/2},b) = O(b^2)$:
  - precompute decision tree in time $2^{O(b^2)} = n^{O(\varepsilon)}$

$$\Rightarrow T(n) \leq O((n/b)\cdot(n/b^{1/2})\cdot(n/b) \cdot b^2)$$
$$= \boxed{O(n^3 / \log^{1/4} n)}$$

Example 3.1: All-Pairs Shortest Paths Alg'm [C., STOC'07]

- $T(n) \leq (n/b_1)(n/b_2)(n/b_3)\,[T(b_1,b_2,b_3) + O(\,b_1 b_3 \qquad)]$
- For $b = w_0/\log w_0$, $T(n,b,n) = O^*(\,n^2 \qquad)$:
  - idea: view as b-dimensional geometric problem!
  - map row i of A to point $p_i = (a_{i1},\ldots,a_{ib})$
  - map column j of B to $O(b^2)$ hyperplanes

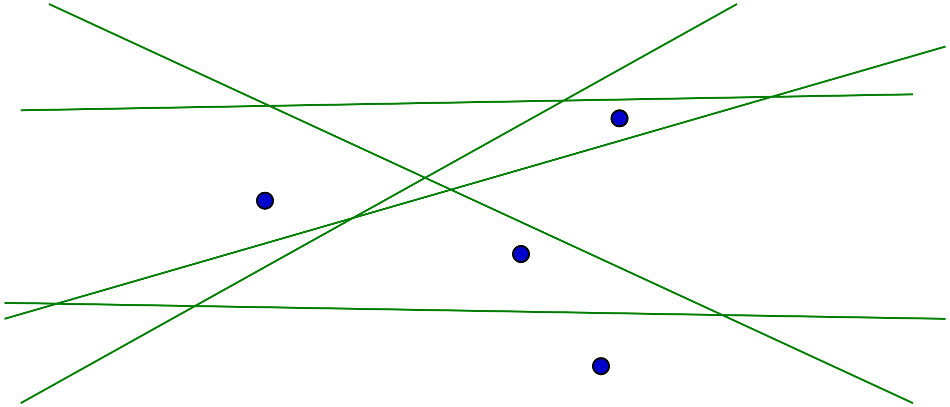
    $h_{jkk'} = \{(x_1,\ldots,x_b) \mid x_k + b_{kj} = x_{k'} + b_{k'j}\}$



Example 3.1: All-Pairs Shortest Paths Alg'm [C., STOC'07]

- $T(n) \leq (n/b_1)(n/b_2)(n/b_3)\,[T(b_1,b_2,b_3) + O(\,b_1b_3 \qquad)]$
- For $b = w_0/\log w_0$, $T(n,b,n) = O^*(\,n^2 \qquad)$:
  - idea: view as b-dimensional geometric problem!
  - map row i of *A* to point $p_i = (a_{i1},\ldots,a_{ib})$
  - map column j of B to $O(b^2)$ hyperplanes

    $h_{jkk'} = \{(x_1,\ldots,x_b) \mid x_k + b_{kj} = x_{k'} + b_{k'j}\}$



Example 3.1: All-Pairs Shortest Paths Alg'm [C., STOC'07]

- $T(n) \le (n/b_1)(n/b_2)(n/b_3)\,[T(b_1,b_2,b_3) + O^*(b_1 b_3/w_0)]$

  tricky!

- For $b = w_0/\log w_0$, $T(n,b,n) = O^*(n^2/w_0)$:
  - idea: view as b-dimensional geometric problem!
  - map row i of A to point $p_i = (a_{i1},\ldots,a_{ib})$
  - map column j of B to $O(b^2)$ hyperplanes

    $h_{jkk'} = \{(x_1,\ldots,x_b) \mid x_k + b_{kj} = x_{k'} + b_{k'j}\}$
  - want to classify each point against each hyperplane
  - subquadratic time by comp. geometry techniques, which work well for dimensions b up to log n/loglog n

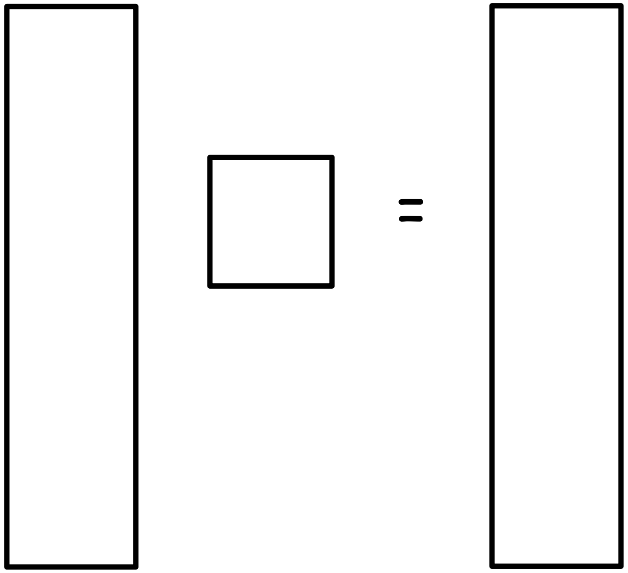
$\Rightarrow T(n) \le O^*((n/b) \cdot n^2) = \boxed{O^*(n^3/\log^2 n)}$

Example 3.2: Exact TSP

- Standard dynamic programming by Held,Karp'62:

$$C[S,j] = \min_k \left(C[S-\{k\},k] + a_{kj}\right) \quad \forall S \subset \{1,\dots,n\},\ j \notin S$$

- This is basically min-plus matrix multiplication $T(2^n,n,n)$!



Example 3.2: Exact TSP

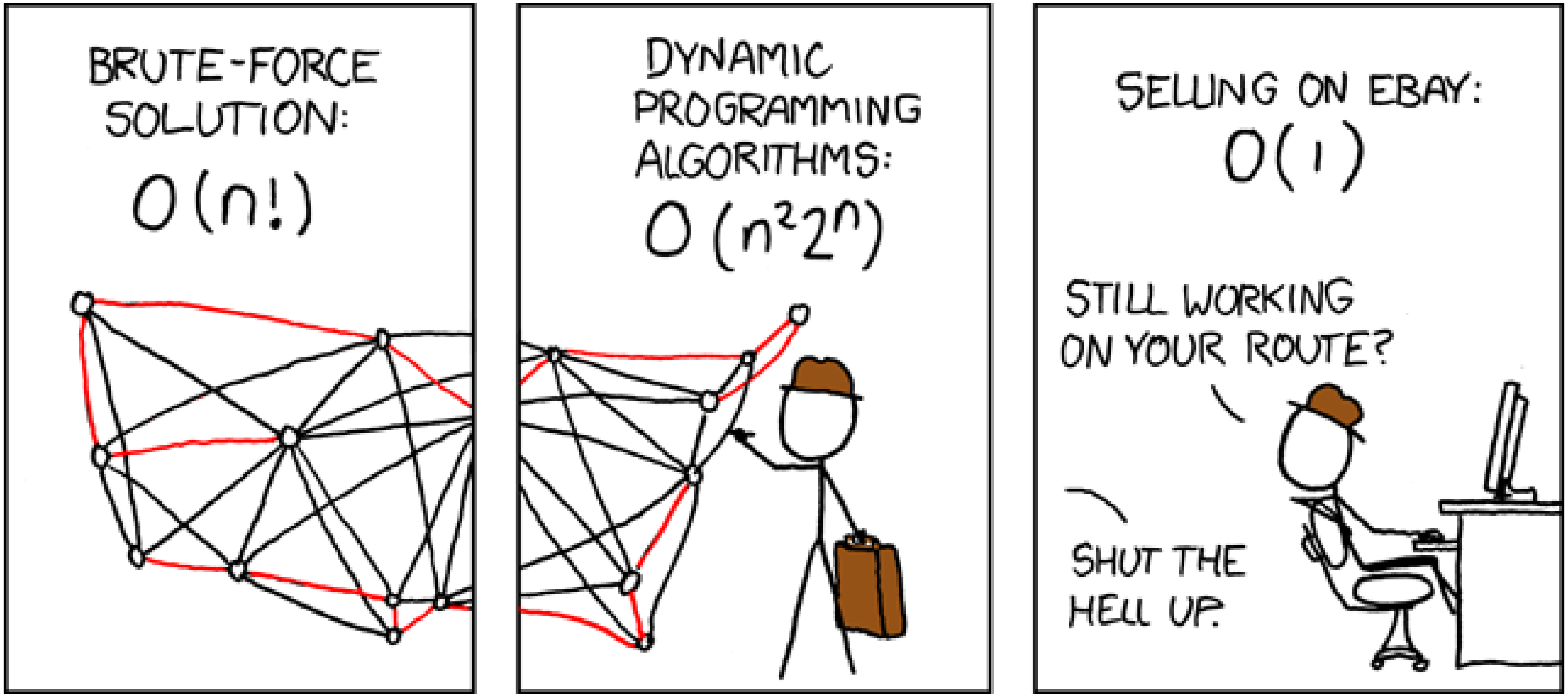
- E.g., use Fredman's approach

- For $b = \varepsilon\, n^{1/2}$, $T(b, b^{1/2}, b) = O(b^2)$:
  - precompute decision tree in time $2^{O(b^2)} \ll 2^n$

$$\Rightarrow T(2^n,n,n) \le O((2^n/b)\cdot(n/b^{1/2})\cdot(n/b) \cdot b^2)$$

$$= \boxed{O(\cancel{n^{1.75}}\, 2^n)} \qquad \text{instead of } O(n^2\, 2^n)$$

$n^{1.5}$ (by another approach)



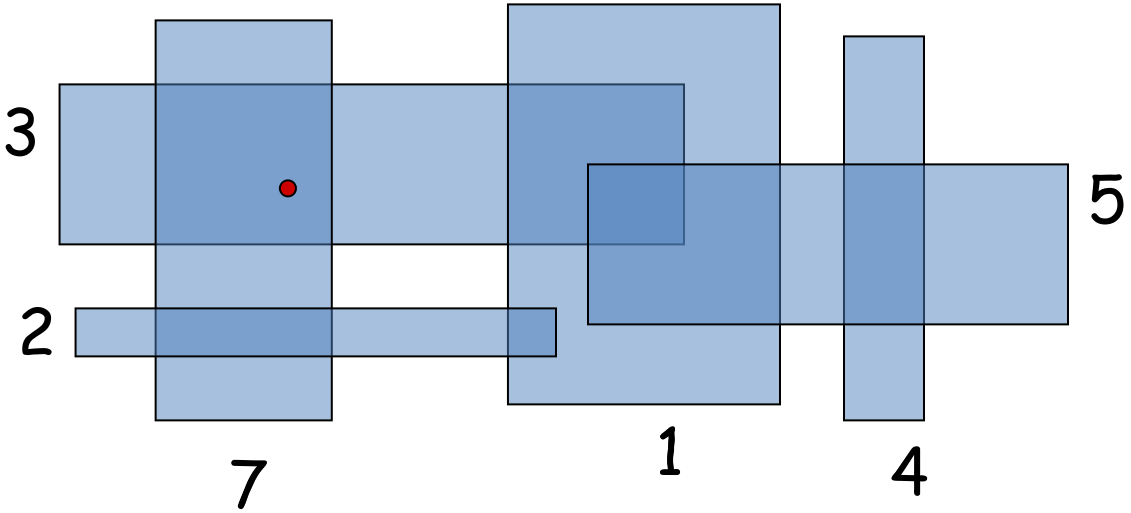
(from http://xkcd.com/399)

Example 3.2: Exact TSP
Remarks

- But issues with the model... (need n-bit words !)
- Decision tree complexity for TSP known to O(n^8 log n) (polynomial !) [Kolinek'87, Meyer auf der Heide'84]

Example 3.3: Weighted Box Depth

- Given n weighted boxes in d ≥ 3 dimensions,
  - find a point with max/min depth
    where depth(p) = sum of weights of boxes containing p



Example 3.3: Weighted Box Depth Decision Tree Complexity

- $T(n) = O(n^5 \log n)$:
  - compute the arrangement of boxes by $O(n \log n)$ comparisons
  - answer is max of $O(n^d)$ linear functions over the $O(n)$ weights
  - idea: view as $O(n)$-dimensional geometric problem !
  - D-dimensional point location for N hyperplanes in $O(D^5 \log N)$ query time [Meiser'93, Meyer auf der Heide'84]

Example 3.3: Weighted Box Depth Alg'm [C.,FOCS'13]

- $T(n) \le O(n/b)^{d/2}\,[T(b) + O(b)]$

- For $b = w_0/\log w_0$, $T(b) = O(b^5 \log b)$:
  - preprocess point location structure, which works well for dimensions b up to log n/loglog n

$\Rightarrow\; T(n) = \boxed{O^*((n/\log n)^{d/2} \log^5 n)}$

Final Open Questions

- kSUM for real numbers in $O(n^{(k+1)/2} / \text{polylog } n)$ time??
  - decision tree complexity known to be $O(n^4 \log n)$ [Meyer auf der Heide'84]
  - but no good divide&conquer for $k > 3$
- d-dimensional affine degeneracy testing for real numbers in $O(n^d / \text{polylog } n)$ time??
  - can do divide&conquer
  - but no good decision tree complexity bounds… yet
- Klee's measure problem for real numbers in $O(n^{d/2} / \text{polylog } n)$ time??
- Speedup beyond log factors?? Lower bounds??